Software Development Methodologies

Lecturer: **Raman Ramsin**

Lecture 7

Agile Methodologies: Scrum

Agile Methodologies: Brief History

- First appeared in 1995.

- The once-common perception that agile methodologies are nothing but controlled code-&-fix approaches, with little or no sign of a clear-cut process, is only true of a small – albeit influential – minority.

- Essentially based on practices of program design, coding and testing that are believed to enhance software development flexibility and productivity.

- Most agile methodologies incorporate explicit processes, although striving to keep them as lightweight as possible.

Agile Methodologies: Evolution Map
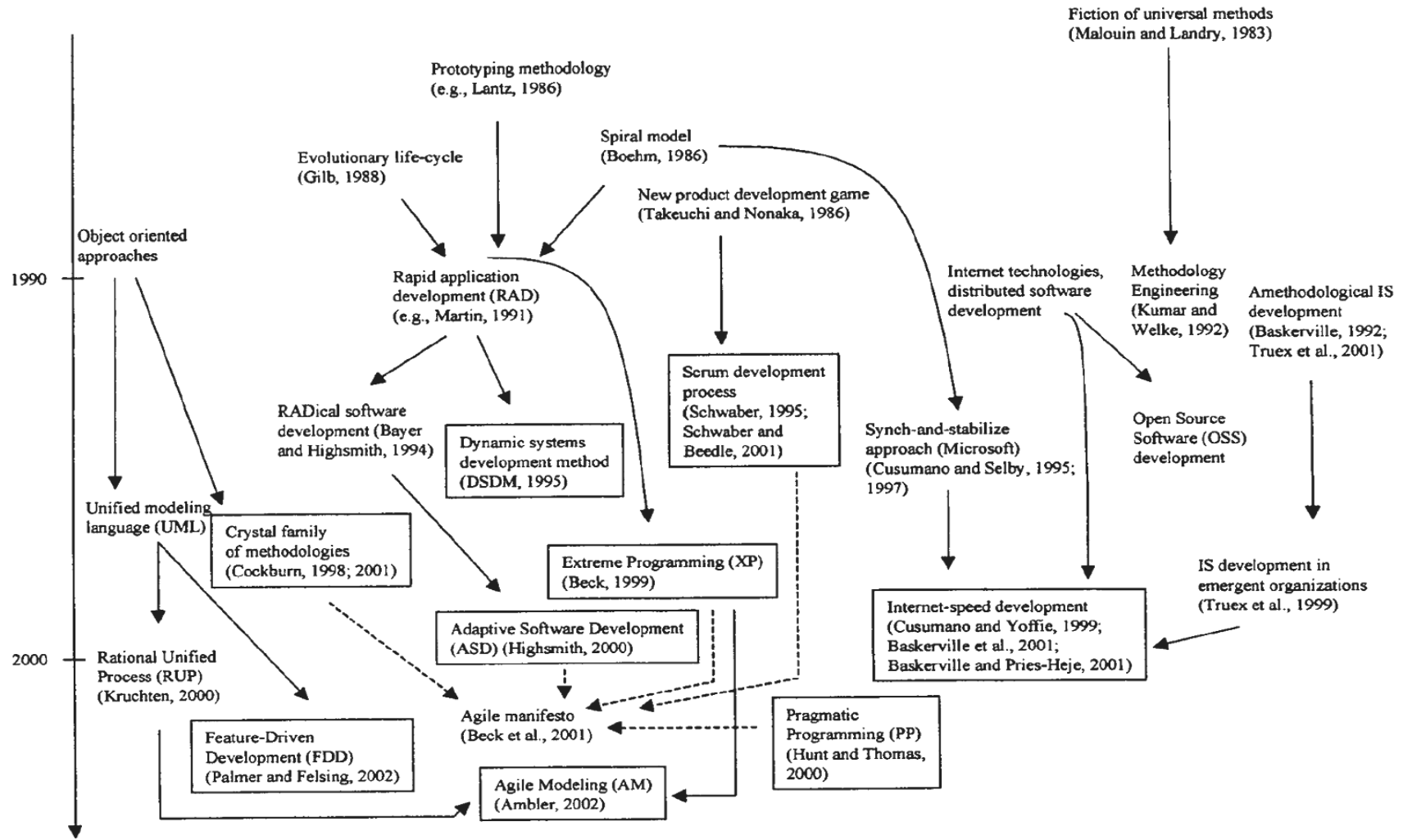
[Abrahamsson et al. 2003]

Agile Methodologies: Agile Manifesto

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Agile Methodologies: Principles

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

- Business people and developers must work together daily throughout the project.

- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Sharif University of Technology

Agile Methodologies: Principles (Contd.)

- Working software is the primary measure of progress.

- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

- Continuous attention to technical excellence and good design enhances agility.

- Simplicity—the art of maximizing the amount of work not done—is essential.

- The best architectures, requirements, and designs emerge from self-organizing teams.

- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

6
Sharif University of Technology

Scrum

- First mentioned as a development method in 1986, referring to a fast and flexible product development process practiced in Japanese manufacturing companies.

- The variant of Scrum used for software development, jointly developed by Sutherland and Schwaber, was introduced in 1995.

- The name emphasizes the importance of teamwork in the methodology and is derived from the game of rugby.

- Originally intended as a general framework for systems development, but is currently used as a comprehensive software development methodology.
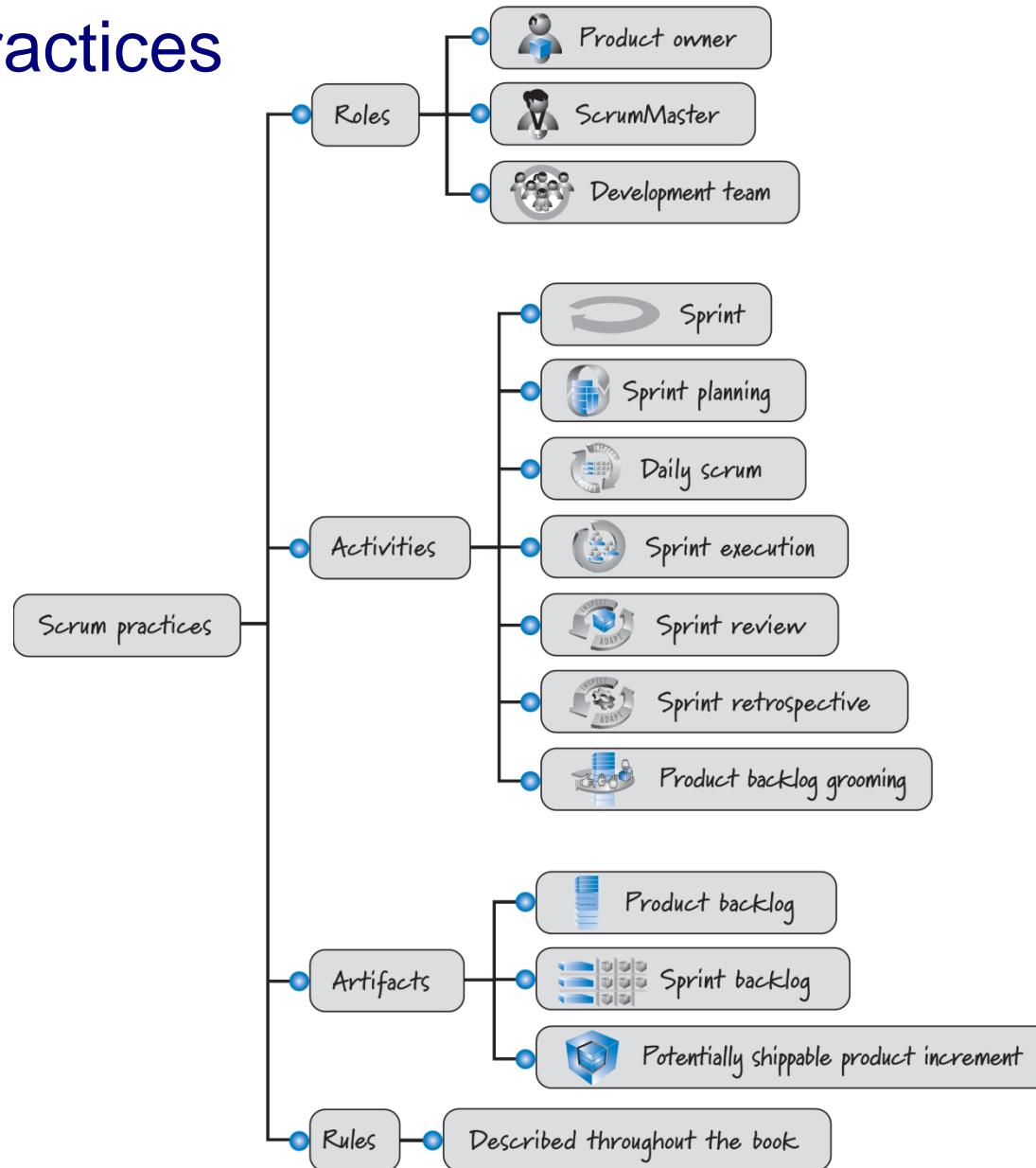
Scrum: Process Framework

1. A people-centric framework based on a set of **values**, **principles**, and **practices** that provide the foundation to which an organization can add its unique implementations for realizing the Scrum practices.

2. Scrum **Values**: Honesty, Openness, Courage, Respect, Focus, Trust, Empowerment, and Collaboration.

3. Scrum **Principles**: Manifestations of the Agile Principles.

4. Scrum **Practices**: Embodied in specific **roles**, **activities**, **artifacts**, and their associated **rules.**

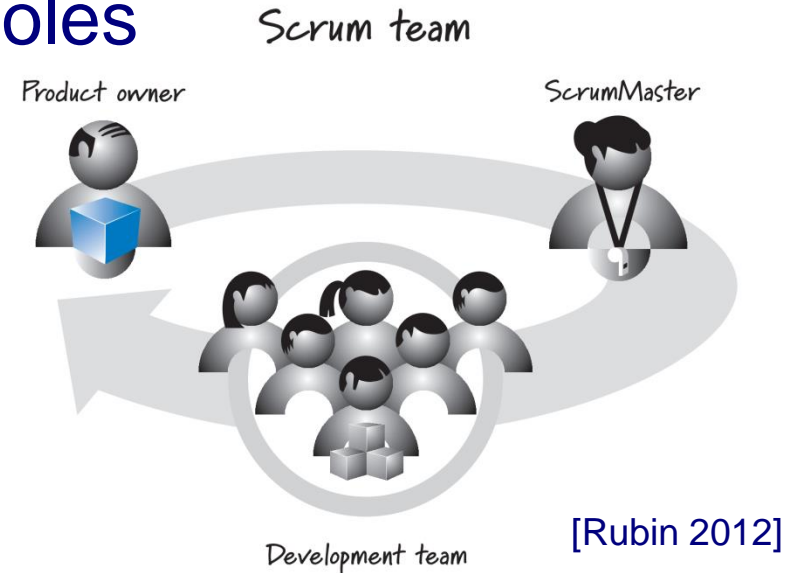5. Introduced herein based on Rubin's book (2012), and The Scrum Guide (2020).

Sharif University of Technology

Scrum Practices

[Rubin 2012]

Scrum Practices: Scrum Team Roles

Scrum team

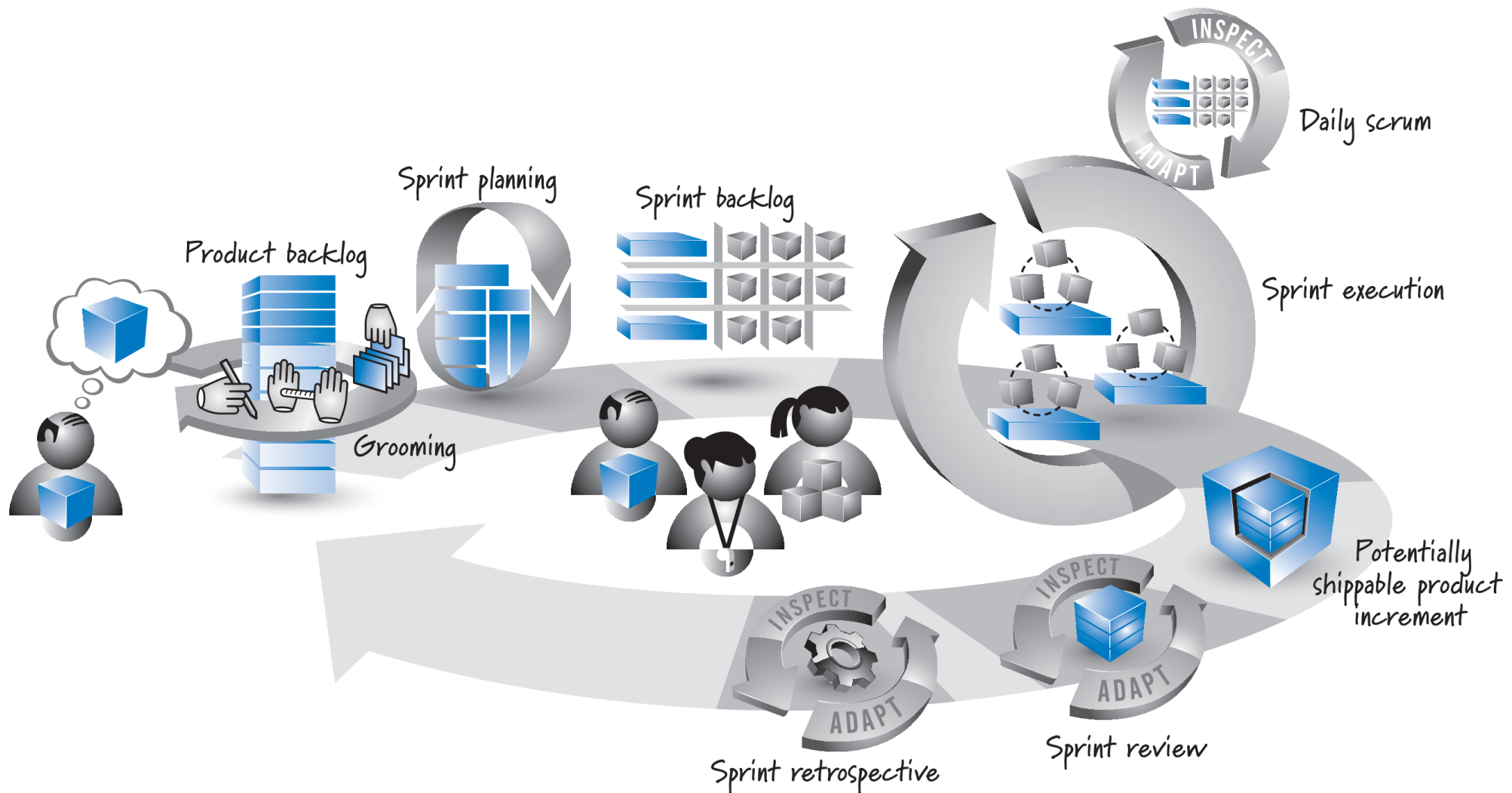Product owner

ScrumMaster

[Rubin 2012]

Development team

1. **Product Owner:** Responsible for what will be developed and in what order.

2. **Scrum Master:** Responsible for guiding the team in creating and following its own process based on the broader Scrum framework.

3. **Development Team (Developers):** Responsible for determining how to deliver what the product owner has asked for.

Scrum Process: Activities and Artifacts

1. **Product owner** has a **vision** of what he wants to create.
   - ☐ Through an activity called **grooming,** the vision is broken down into a set of features that are collected into a prioritized list called the **product backlog**.
2. **Sprints** are performed iteratively; each sprint consists of:
   1. **Sprint planning:** At the beginning of each sprint:
      1. The development team selects a subset of the product backlog items (**features**) it believes it can commit to completing.
      2. A **sprint backlog** is created; it describes, through a set of detailed **tasks**, how the team plans to design/build/integrate/test the selected features.
   2. **Sprint execution:** The development team performs the tasks necessary to realize the selected features.
      1. Each day, the Scrum team (ten members or less) conduct a synchronization, inspection, an adaptive planning activity known as the **daily scrum**.
      2. At the end of execution, the team has produced a **potentially shippable product increment** that represents some of the product owner's vision.
   3. **Sprint review:** Stakeholders and Scrum team inspect and adapt the product being built.
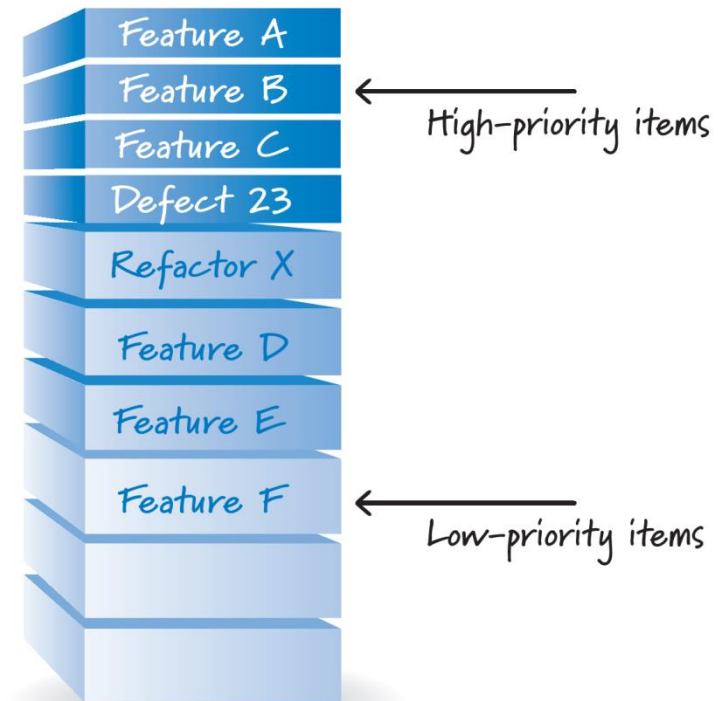   4. **Sprint retrospective:** Scrum team inspects and adapts the Scrum process being used to create the product.

11

Sharif University of Technology

# Scrum Process: Activities and Artifacts

[Rubin 2012]

Product Backlog

1. The product owner, with input from the rest of the Scrum team and stakeholders, is responsible for determining and managing the sequence of work in the form of the product backlog.

   □ Initially, product backlog items are features required to meet the product owner's vision.

   □ During development, the backlog also contains new features, changes to existing features, defects needing repair, and technical improvements.

2. The product owner collaborates with internal and external stakeholders to gather and define the product backlog items.

   □ High-value items appear at the top of the product backlog and the lower-value items appear toward the bottom.

Feature A
Feature B   ← High-priority items
Feature C
Defect 23
Refactor X
Feature D
Feature E
Feature F   ← Low-priority items

[Rubin 2012]

Department of Computer Engineering

13

Sharif University of Technology

Product Backlog: Grooming

1. Overall, the activity of creating and refining product backlog items, estimating them, and prioritizing them is known as **grooming**.

2. Product backlog items are placed in the correct sequence using factors such as **value**, **cost**, **knowledge**, and **risk**.

3. Prioritization requires **estimation** of the size of each product backlog item.

    □ Size equates to cost.

    □ Scrum does not dictate which size measure to use.

    □ Relative size measures are usually used; such as **story points** or **ideal days**.

      ■ Instead of the absolute value, the relative size of an item compared to other items is considered.

[Rubin 2012]
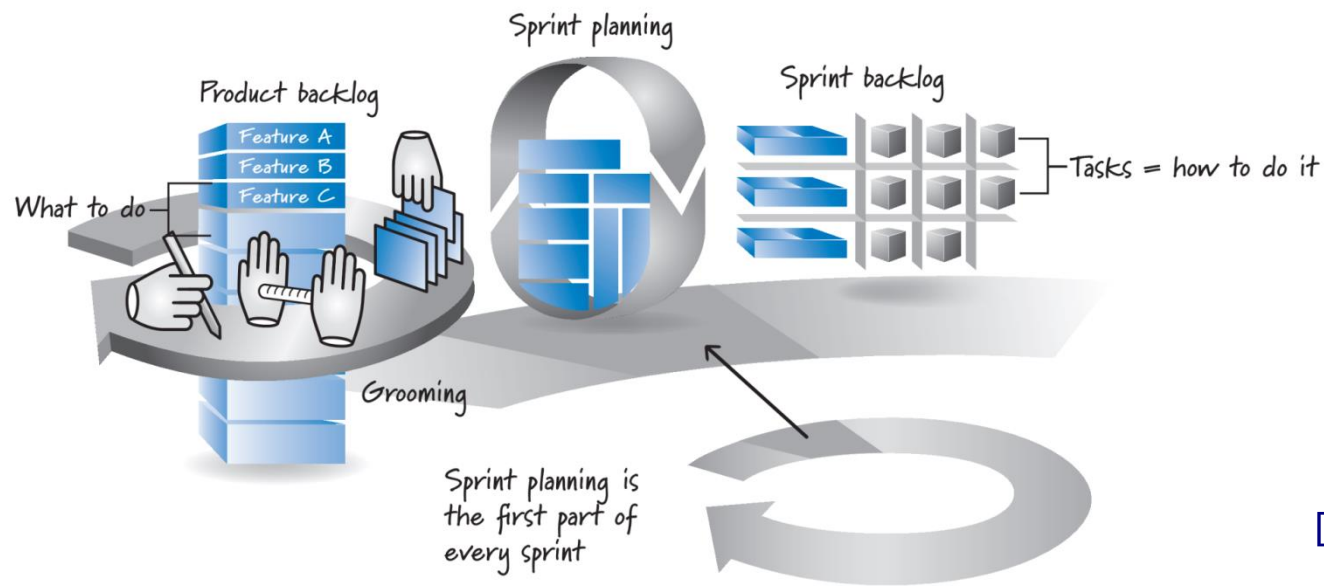
Sharif University of Technology
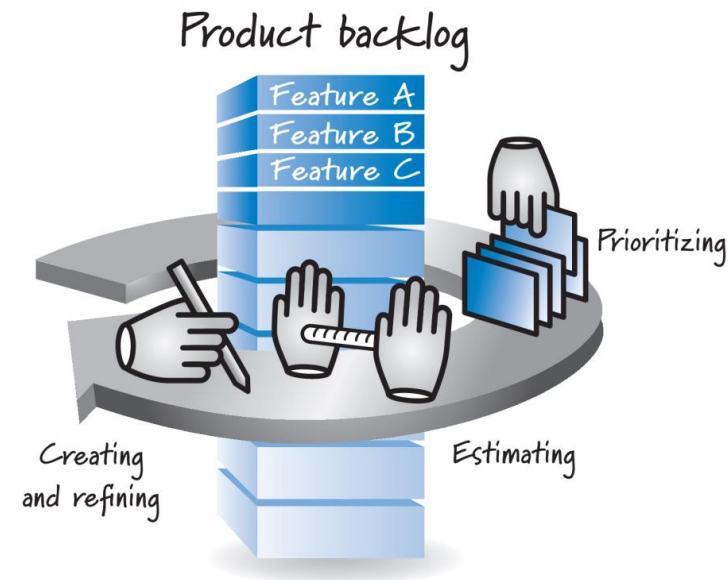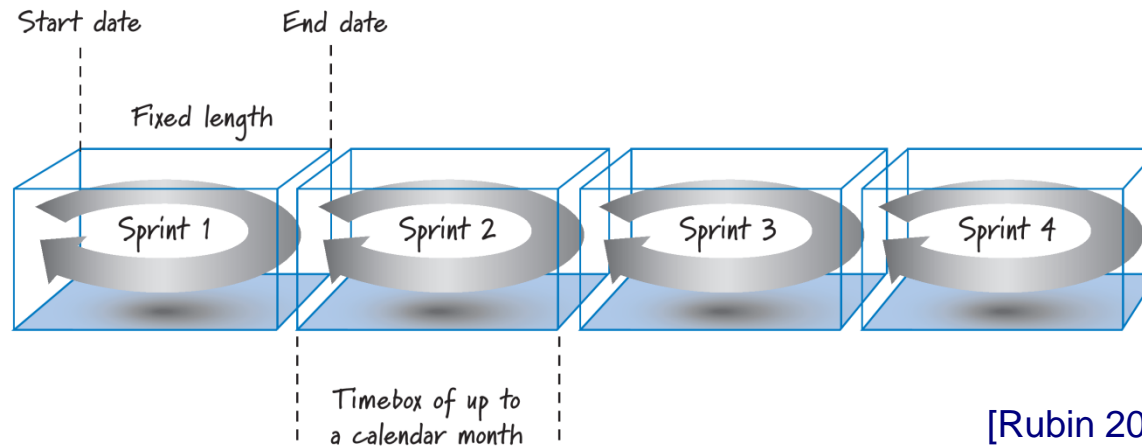
Sprints

[Rubin 2012]

1. In Scrum, work is performed in iterations or cycles of up to a calendar month called sprints.

2. The work completed in each sprint should create something of tangible value to the customer or user.

3. Sprints are timeboxed so they always have a fixed start and end date, and generally they should all be of the same duration.

4. As a rule we do not permit any goal-altering changes in scope or personnel during a sprint, unless absolutely necessary.

Sprint Planning

Sprint planning

Product backlog

Sprint backlog

Feature A
Feature B
Feature C

What to do

Tasks = how to do it

Grooming

Sprint planning is the first part of every sprint

[Rubin 2012]

1. To determine the most important subset of product backlog items to build in the next sprint, the Scrum team performs sprint planning.

2. During sprint planning, the product owner and development team agree on a **sprint goal** for the upcoming sprint.

   □ Using this goal, the development team determines the high-priority product backlog items for the upcoming sprint.
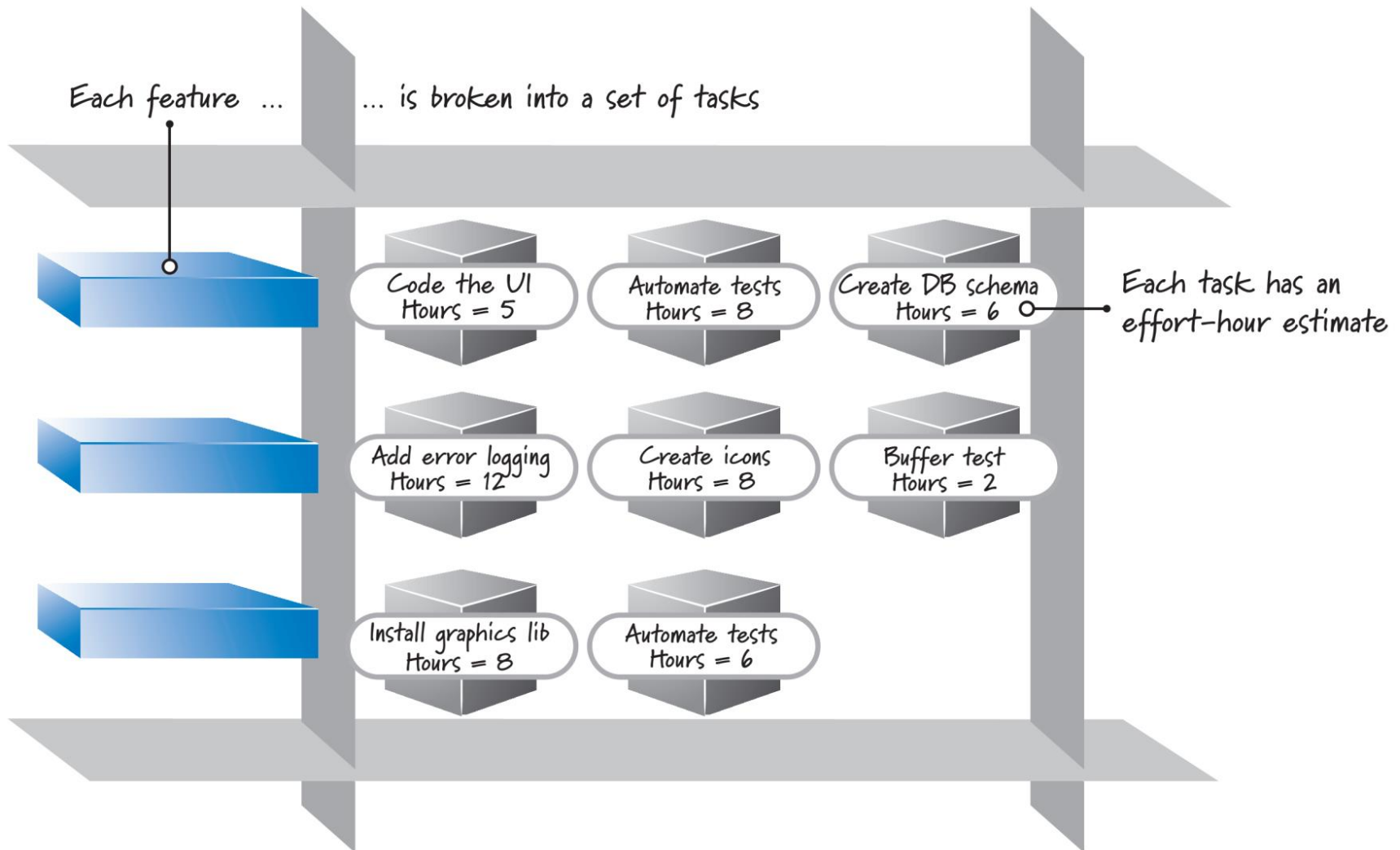
Sprint Planning: Sprint Backlog

1. The development team breaks down each targeted feature into a set of tasks.

   The collection of these tasks, along with their associated product backlog items, forms a second backlog called the sprint backlog.

2. The development team then provides an estimate (typically in hours) of the effort required to complete each task.

3. There are several approaches that can be used for sprint planning. The preferred approach is as follows:

   1. Select a product backlog item;

   2. break the item down into tasks, and determine if the selected item will reasonably fit within the sprint;

   3. If it does fit and there is more capacity to complete work, repeat the cycle until the team is out of capacity to do any more work.

Sprint Planning: Sprint Backlog

Each feature ...   ... is broken into a set of tasks

Code the UI
Hours = 5

Automate tests
Hours = 8

Create DB schema
Hours = 6

Each task has an
effort-hour estimate

Add error logging
Hours = 12

Create icons
Hours = 8

Buffer test
Hours = 2

Install graphics lib
Hours = 8

Automate tests
Hours = 6

[Rubin 2012]

Sprint Execution

1. The development team, guided by the Scrum Master's coaching, performs all the task-level work necessary to get the features done.

   ☐ "Done" means there is a high degree of confidence that all of the work necessary for producing good-quality features has been completed.

2. Exactly what tasks the team performs depends on the nature of the work.

   ☐ For example, are we building software and what type of software, or are we building hardware, or is this marketing work?

3. Nobody tells the development team in what order or how to do the task-level work in the sprint backlog.

   ☐ Team members define their own task-level work and then self-organize in any manner they feel is best for achieving the sprint goal.

Sharif University of Technology

Sprint Execution: Daily Scrum

1. Each day of the sprint, ideally at the same time, the development team members hold a timeboxed (15 minutes or less) daily scrum.

2. The Scrum Master facilitates the meeting and each team member answers three questions for the benefit of the other team members:

    1. What did I accomplish since the last daily scrum?
    2. What do I plan to work on by the next daily scrum?
    3. What are the obstacles that are preventing me from making progress?

3. By answering these questions, everyone understands:

    1. The big picture of what is occurring;
    2. How they are progressing toward the sprint goal;
    3. Any modifications they want to make to their plans for the upcoming day's work; and
    4. What issues need to be addressed.

Sharif University of Technology

Sprint Execution: Rules of Daily Scrum
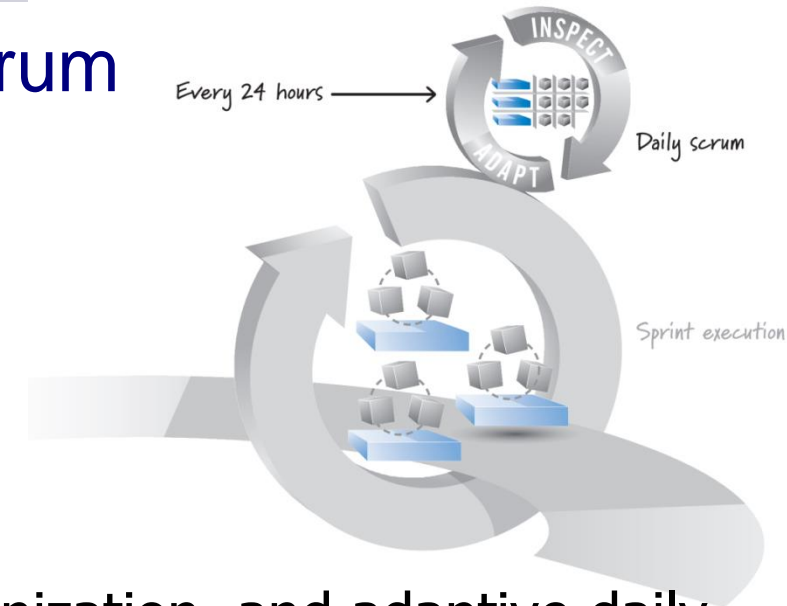
1. The daily scrum is an inspection, synchronization, and adaptive daily planning activity that helps a self-organizing team do its job better.

2. The daily scrum is not a problem-solving activity.

   □ Rather, many teams decide to talk about problems after the daily scrum and do so with a small group of interested people.

3. The daily scrum is not a traditional status meeting.

4. At the daily scrum, only the **pigs** should talk; the **chickens**, if any, should attend as observers.

Sprint Execution: Potentially Shippable Product Increment

1. In Scrum, we refer to the sprint results as a **potentially shippable product increment**.

2. Whatever the Scrum team agreed to do should be really done according to its agreed-upon **definition of done**.

3. "Potentially shippable" does not mean that what got built must actually be shipped.

   □ Shipping is a business decision, frequently influenced by things such as:

      ▪ "Do we have enough features to justify a deployment?" or

      ▪ "Can our customers absorb another change given that the last release was made just two weeks ago?"

4. Potentially shippable is better thought of as a state of confidence that there is no materially important work left undone.

Sharif University of Technology

Sprint Review

1. The goal is to inspect and adapt the **product** that is being built.

2. Participants include the Scrum team, stakeholders, sponsors, customers, and interested members of other teams.

3. Conversation is focused on reviewing the just-completed features in the context of the overall development effort.

4. Everyone in attendance gets clear visibility into what is occurring and has an opportunity to help guide the forthcoming development.

5. Bidirectional information flow:

   □ The people who are not on the Scrum team get to sync up on the development effort and help guide its direction.

   □ Scrum team members gain a deeper appreciation for the business and marketing side of their product.

Sprint Retrospective

1. An opportunity to inspect and adapt the Scrum **process**.

2. The development team, Scrum Master, and product owner meet to discuss what is and is not working with their Scrum process and its associated practices.

3. The focus is on continuous process improvement.
   - The Scrum team identifies and commits to a practical number of process improvement actions, to be undertaken by the team in the next sprint.

Scrum: Strengths and Weaknesses

- ## *<u>Strengths</u>*

  - ☐ Iterative-incremental process

  - ☐ Requirements are allowed to evolve over time.

  - ☐ Traceability to requirements through the *Product Backlog*

  - ☐ Iterative development engine governed by careful planning and reviewing

  - ☐ Active user involvement

  - ☐ Simple and straightforward process

  - ☐ Early and frequent releases, demonstrating functionality at the end of each iteration (*sprint*) of the development cycle

Scrum: Strengths and Weaknesses

- ***Weaknesses***

  - Lack of scalability

  - Based on the assumption that human communication is sufficient for running projects of any size and keeping them focused

  - Not necessarily seamless

  - No clear-cut design effort

  - Model-phobic

  - Lack of formalism

Sharif University of Technology

References

- Schwaber, K., Beedle, M., *Agile Software Development with Scrum.* Prentice-Hall, 2001.

- Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J., *Agile Software Development Methods: Review and Analysis*. VTT Publications, 2002.

- Rubin, K.S., *Essential Scrum: A Practical Guide to the Most Popular Agile Process*, Addison-Wesley, 2012.

- Schwaber, K., Sutherland, J., *The Scrum Guide*, Published online at: http://www.scrumguides.org/, November 2020 (last visited on: 27 December 2020).